

AfxLoadLibrary

Vulnerable to "tainted" DLLs placed in a location in the search path before the intended DLL

Sean Barnum, Cigital, Inc. [vita¹]

Copyright © 2005 Cigital, Inc.

2005-10-03

Part "Original Cigital Coding Rule in XML"

Mime-type: text/xml, size: 5661 bytes

Attack Category	<ul style="list-style-type: none">• Resource Injection
Vulnerability Category	<ul style="list-style-type: none">• Indeterminate File/Path
Software Context	<ul style="list-style-type: none">• Process Management• File Path Management
Location	
Description	<p>AfxLoadLibrary() and CoLoadLibrary() have many implicit, default behaviors that can give an attacker an opportunity to inject object code into your code base.</p> <p>The AfxLoadLibrary() function is used to load code from a DLL library. The system will use ".DLL" for the file extension if it is not specified. If the path for the library to load is not fully qualified, then the system will search the following locations in this order:</p> <ol style="list-style-type: none">1. The directory from which the application loaded.2. The current working directory. (Could be affected by chdir())3. The Windows system or system32 directory. On Windows 95/98/Me this is the Windows system directory. On more recent versions of windows it is system32.4. Windows NT only: The 16-bit Windows system directory. There is no Win32 function that obtains the path of this directory, but it is searched. The default name of this directory is SYSTEM.5. The Windows directory (typically C:\WINDOWS or C:\WINNT).6. Each directory listed in the PATH environment variable, in the order they are listed. <p>A somewhat different search strategy is used for CoLoadLibrary().</p>

1. <http://buildsecurityin.us-cert.gov/bsi-rules/35-BSI.html> (Barnum, Sean)

	This issue has reportedly been fixed in Windows XPSP1, Windows Server 2003 and newer versions.		
APIs	FunctionName		Comments
	AfxLoadLibrary		
	CoLoadLibrary		
	CoLoadLibrary		
Method of Attack	An attacker could inject a Trojan horse DLL within your process by placing a "tainted" DLL in a location in the DLL search path that is found before the intended DLL.		
Exception Criteria			
Solutions	Solution Applicability	Solution Description	Solution Efficacy
	When library is loaded.	The lpzModuleName or lpzLibName parameter should be a fully qualified filename, including the file extension.	Effective if identified file is secure from tampering.
Signature Details	Definite positive signature: call to AfxLoadLibrary() with a literal string that does not begin with "\\\" or "[A-Z]:\" Possible positive signature: call to AfxLoadLibrary() with a string or character array variable Definite negative signature:call to AfxLoadLibrary() with a literal string that begins with a drive letter or UNC path (e.g. "\\\"")		
Examples of Incorrect Code	AfxLoadLibrary("MyLib"); // or AfxLoadLibrary("Prog\MyLib.DLL");		
Examples of Corrected Code	AfxLoadLibrary("C:\\Program Files\\MyCompany\\MyProg\\MyLib.DLL"); // or AfxLoadLibrary("\\Server\\Share\\Path\\MyLib.DLL");		
Source References			
Recommended Resources	<ul style="list-style-type: none">• MSDN reference for AfxLoadLibrary²• MSDN reference for CoLoadLibrary³		
Discriminant Set	Operating Systems	<ul style="list-style-type: none">• Windows 98• Windows Me• Windows 2000	

		<ul style="list-style-type: none"> • Windows XP Home • Windows XP Pro • Win32
	Languages	<ul style="list-style-type: none"> • C • C++

Cigital, Inc. Copyright

Copyright © Cigital, Inc. 2005-2007. Cigital retains copyrights to this material.

Permission to reproduce this document and to prepare derivative works from this document for internal use is granted, provided the copyright and “No Warranty” statements are included with all reproductions and derivative works.

For information regarding external or commercial use of copyrighted materials owned by Cigital, including information about “Fair Use,” contact Cigital at copyright@cigital.com¹.

The Build Security In (BSI) portal is sponsored by the U.S. Department of Homeland Security (DHS), National Cyber Security Division. The Software Engineering Institute (SEI) develops and operates BSI. DHS funding supports the publishing of all site content.

1. <mailto:copyright@cigital.com>